

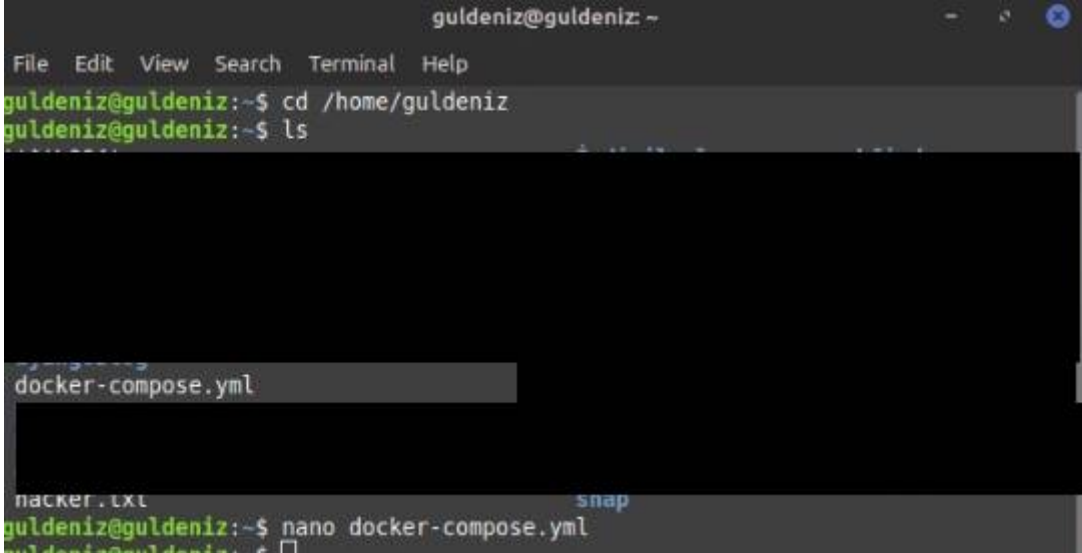
# Kurulum

TheHive'ı Docker üzerinde kurmak için, Docker'ın kurulu olduğundan ve çalıştığından emin olun. Daha sonra, Docker komut satırında uygun TheHive Docker imajını indirin ve çalıştırın. Bu, TheHive'ın hızlı bir şekilde başlatılmasını ve herhangi bir karmaşıklık olmadan çalışmasını sağlar.

- [Docker ile Kurulum](#)

# Docker ile Kurulum

Docker compose dosyasını kullanarak TheHive ve Cortex'i çalıştırın:

A terminal window titled 'guldeniz@guldeniz: ~' showing the following commands and output:

```
File Edit View Search Terminal Help
guldeniz@guldeniz:~$ cd /home/guldeniz
guldeniz@guldeniz:~$ ls
...
docker-compose.yml
...
guldeniz@guldeniz:~$ nano docker-compose.yml
```

Compose dosyasına aşağıdaki kodu ekleyin.

```
version: "3"
services:
  thehive:
    image: strangebee/thehive:5.2
    depends_on:
      - cassandra
      - elasticsearch
      - minio
      - cortex
    mem_limit: 1500m
    ports:
      - "9000:9000"
    environment:
      - JVM_OPTS="-Xms1024M -Xmx1024M"
    command:
      - --secret
      - "mySecretForTheHive"
      - "--cql-hostnames"
      - "cassandra"
```

- "--index-backend"
- "elasticsearch"
- "--es-hostnames"
- "elasticsearch"
- "--s3-endpoint"
- "http://minio:9000"
- "--s3-access-key"
- "minioadmin"
- "--s3-secret-key"
- "minioadmin"
- "--s3-bucket"
- "thehive"
- "--s3-use-path-access-style"
- "--cortex-hostnames"
- "cortex"
- "--cortex-keys"

# put cortex api key once cortex is bootstrapped

- "<cortex\_api\_key>"

cassandra:

image: 'cassandra:4'

mem\_limit: 1600m

ports:

- "9042:9042"

environment:

- MAX\_HEAP\_SIZE=1024M
- HEAP\_NEWSIZE=1024M
- CASSANDRA\_CLUSTER\_NAME=TheHive

volumes:

- cassandradata:/var/lib/cassandra

restart: on-failure

elasticsearch:

image: docker.elastic.co/elasticsearch/elasticsearch:7.17.12

mem\_limit: 1500m

ports:

- "9200:9200"

environment:

- discovery.type=single-node
- xpack.security.enabled=false

volumes:

- elasticsearchdata:/usr/share/elasticsearch/data

minio:

image: quay.io/minio/minio

mem\_limit: 512m

command: ["minio", "server", "/data", "--console-address", ":9090"]

environment:

- MINIO\_ROOT\_USER=minioadmin
- MINIO\_ROOT\_PASSWORD=minioadmin

ports:

- "9090:9090"

volumes:

- "miniodata:/data"

cortex:

image: thehiveproject/cortex:3.1.7

depends\_on:

- elasticsearch

environment:

- job\_directory=/tmp/cortex-jobs

volumes:

- /var/run/docker.sock:/var/run/docker.sock
- /tmp/cortex-jobs:/tmp/cortex-jobs

ports:

- "9001:9001"

volumes:

miniodata:

cassandradata:

elasticsearchdata:



```
environment:
  - JVM_OPTS="-Xms1024M -Xmx1024M"
volumes:
  - <host_conf_folder>:/data/conf
command:
  - --no-config
  - --config-file
  - /data/conf/application.conf
```

...

```
docker run --rm -p 9000:9000 -v <host_conf_folder>:/data/conf strangebee/thehive:<version> --no-config --
config-file /data/conf/application.conf
```

<host\_conf\_folder> klasörünün bir application.conf dosyası içermesi gerekmektedir.

--no-config, giriş noktasına herhangi bir yapılandırma dosyası oluşturulmamasını söylemek için kullanılır. Aksi takdirde, giriş noktası varsayılan bir yapılandırma oluşturacak ve bu dosya sizin dosyanızla birleştirilecektir.

## Komut Satırı Argümanları Kullanarak

Veri depolama için Cassandra ve Elasticsearch ile TheHive'ı çalıştırmanızı ve dosya depolama için minio'yu kullanmanızı öneririz. Örnek olarak, örneklerinizin ana bilgisayar adlarını argümanlar aracılığıyla iletebilirsiniz:

```
docker run --rm -p 9000:9000 strangebee/thehive:<version> \
  --secret <secret>
  --cql-hostnames <cqlhost1>,<cqlhost2>,...
  --cql-username <cqlusername>
  --cql-password <cqlpassword>
  --index-backend elasticsearch
  --es-hostnames <eshost1>,<eshost2>,...
  --s3-endpoint <minio_endpoint>
  --s3-access-key <minio_access_key>
  --s3-secret-key <minio_secret_key>
```

Bu, docker konteynerinizi harici cassandra ve elasticsearch düğümlerine bağlayacaktır. Veri dosyaları minio üzerinde saklanacaktır. Konteyner, TheHive'ı 9000 numaralı bağlantı noktasında gösterir.

## Tüm Seçenekler

Docker giriş noktası tarafından desteklenen tüm seçenekleri -h ile alabilirsiniz:

```
docker run --rm strangebee/thehive:<version> -h
```

Available options:

--config-file <file>	Yapılandırma dosyasının yolunu belirtir.
--no-config	TheHive'in gizli diziler ve Elasticsearch ayarları eklemek de dahil olmak üzere kendini yapılandırmaya çalışmasını engeller.
--no-config-secret	Yapılandırmada rastgele oluşturulmuş bir gizli anahtar eklenmesini hariç tutar.
--secret <secret>	Oturumları güvence altına almak için kullanılan gizli anahtar ayarlar.
--show-secret	Oluşturulan gizli anahtar görüntüler.
--no-config-db	Veritabanının otomatik olarak yapılandırılmasını sağlar.
--cql-hostnames <host>,<host>,...	Cassandra örneklerini bulmak için bu ana bilgisayar adlarını çözümler.
--cql-username <username>	Cassandra veritabanı için kullanıcı adını belirtir
--cql-password <password>	Cassandra veritabanı için parolayı belirtir.
--no-cql-wait	Cassandra'nın kullanılabilir olmasını beklemeyi atlar.
--bdb-directory <path>	Cassandra kullanılmıyorsa yerel veritabanının konumunu tanımlar (varsayılan: /data/db).
--index-backend	Dizin için kullanılacak arka ucu belirtir, 'lucene' veya 'elasticsearch' (varsayılan: lucene).
--es-hostnames	Dizin için kullanılan Elasticsearch örneklerini belirtir
--es-index	Kullanılacak Elasticsearch dizin adını belirtir (varsayılan: thehive).
--no-config-storage	Depolamanın otomatik yapılandırılmasını devre dışı bırakır.
--storage-directory <path>	S3 kullanılmıyorsa yerel depolama alanının konumunu belirtir (varsayılan: /data/files.).
--s3-endpoint <endpoint>	AWS S3 için 's3.amazonaws.com' ile S3'ün veya kullanılıyorsa diğer nesne depolamanın uç noktasını belirtir
--s3-region <region>	S3 bölgesini belirtir, MinIO için isteğe bağlıdır.
--s3-bucket <bucket>	Kullanılacak kovanın adını belirtir (varsayılan: thehive), bu kovanın zaten var olması gerekir.
--s3-access-key <key>	S3 erişim anahtarını belirtir (S3 için gereklidir).
--s3-secret-key <key>	S3 gizli anahtarını belirtir (S3 için gereklidir).
--s3-use-path-access-style	MinIO veya başka bir AWS dışı S3 sağlayıcısı kullanılıyorsa bu bayrağı ayarlar, varsayılan olarak sanal ana bilgisayar stili kullanılır.
--no-config-cortex	Cortex yapılandırmasını hariç tutar.
--cortex-proto <proto>	Cortex'e bağlanmak için protokolü tanımlar (varsayılan: http).
--cortex-port <port>	Cortex'e bağlanmak için portu tanımlar (varsayılan: 9001).
--cortex-hostnames <host>,<host>,...	Cortex örneklerini bulmak için bu ana bilgisayar adlarını

çözümler.

--cortex-keys <key>,<key>,... | Cortex anahtarlarını tanımlar.

--kubernetes | Diğer düğümlere katılmak için Kubernetes API'sini kullanır.

• `--kubernetes-pod-label-selector <selector>`: |Uygulamayı çalıştıran diğer podları seçmek için kullanılacak seçiciyi belirtir (varsayılan app=thehive).

`--cluster-min-nodes-count <count>` | Bir küme oluşturmak için minimum düğüm sayısını belirtir (varsayılan 1'dir).

`migrate <param> <param> ...` | Geçiş aracını çalıştırır.

`cloner <param> <param> ...` | Klonlama aracını çalıştırır.

## Kubernetes'te Kullanım

Bir konteyner, Docker'ın izin verdiğiinden çok daha fazla bellek kullanıyor. Konteyner için daha fazla bellek izni vermek için `mem_limit` parametresini artırabilirsiniz. JVM tabanlı uygulamalar için (TheHive, Cassandra, Elasticsearch gibi) JVM parametrelerini ayarlayarak kullanılacak maksimum heap boyutunu belirleyebilirsiniz. TheHive için `JVM_OPTS` ortam değişkenini kullanabilirsiniz:  
`JVM_OPTS="-Xms1024M -Xmx1024M"`